

CONFIGURING NETWORK SETTINGS USING PORTABLE STORAGE MEDIA

[0001] This patent application claims the benefit of U.S. Provisional Patent Application No. 60/534,795 file January 7, 2004.

FIELD OF THE INVENTION

[0002] This invention pertains generally to the field of computer networks and more particularly to a mechanism for simplifying the process of configuring nodes in computer networks.

BACKGROUND OF THE INVENTION

[0003] The use of data communication networks continues to grow. In small as well as large corporate settings, wired local area networks (LANs) and wide area networks (WANs) have become an established feature of conducting business, and wireless networks are being increasingly employed. The use of network technology in the home, both wired and wireless, is a more recent phenomenon and has been slower to develop. In addition to facilitating Internet connectivity, home networking permits personal computing devices and various consumer electronic devices and appliances within the home to communicate with each other. Wireless technology, such as IEEE 802.11 wireless networks and networks of Bluetooth-enabled devices, is attractive in home as well as corporate environments for reasons of convenience, mobility and flexibility.

[0004] A principal impediment to the wider adoption of networking technology in the home and other non-corporate environments has been the difficulty experienced by non-expert users in configuring network devices. For example, it can be troublesome for such users to configure a device to function as a wireless access point (WAP). Setting up an ad hoc wireless network of peer devices (such as wireless PCs, wireless printers, and PDAs) is also typically a complex task. Each peer device must have the same network settings, and a secure ad hoc network typically requires each peer device to have a common WEP key, which must be communicated to the user of the peer device and entered manually. Thin client devices, such as digital audio

receivers and wireless printers, which comprise an increasing number of home network devices, are particularly laborious to configure for network connectivity because they lack the convenient and intuitive I/O capabilities of conventional personal computers.

BRIEF SUMMARY OF THE INVENTION

[0005] In accordance with the invention, a portable computer-readable media device is used to configure network devices to allow network functionality and connectivity. In an embodiment of the invention, a configuration application generates, or aids the user in generating, network settings such as wireless configuration settings, LAN settings, and WAN settings. These settings are determined from the operating system or through user input. The configuration application then generates an Extensible Markup Language (XML) file embodying the network settings and writes that file to the portable computer-readable media device. A user may then install the portable computer-readable media device in one or more other network devices to automatically transfer the configuration to those network devices. In one exemplary embodiment of the invention, a wireless device may be provisioned with the wireless configuration settings necessary to join a wireless network, without requiring the user to manually enter the network settings.

[0006] In a further embodiment of the invention, a method of provisioning a network device with network settings comprises the steps of determining a network settings configuration for allowing the network device to join a network, wherein the network settings configuration includes at least a network name and a network encryption key; generating an XML file including the network settings configuration; and writing the XML file to a portable computer-readable media device. In one permutation of this embodiment, the network encryption key is automatically generated so that a user does not have to enter a lengthy encryption key manually through a user interface.

[0007] As noted above, the network settings configuration is determined in an embodiment of the invention by collecting data from a user through a user interface. In another embodiment of the invention, the network settings configuration is determined at least in part by using an application program interface (API) of an operating system to determine network settings such as wireless configuration, LAN settings, and Internet Protocol (IP) settings.

[0008] In another embodiment of the invention, data stored on a portable computer-readable media device for provisioning a device with network settings comprises an XML file containing a network settings configuration for configuring the device to join a network, and an autorun file for prompting the device to apply the network settings configuration. In this embodiment, the network settings include at least one of (1) wireless configuration settings, (2) LAN settings, (3) WAN settings, and (4) broadband modem settings. The data stored on the portable computer-readable media device can also include multiple versions of the network settings configuration, corresponding to different networks. Still further, the data stored on the portable computer-readable media device can include a device configuration file describing configured devices.

[0009] In keeping with the invention, the data stored on the portable computer-readable media device further includes an application for configuring the portable computer-readable media device, wherein the application executes on a computer to which the portable computer-readable media device is connected. In another embodiment of the invention, a method of configuring a device for operation in a network comprises detecting the installation of a portable computer-readable media device; uploading a configuration from the portable computer-readable media device, wherein the configuration includes network settings, device configuration information, security information, and file sharing information; applying the configuration; and joining the network. In this embodiment, the configuration is again embodied in an XML file stored on the portable computer-readable media device.

[0010] Additional features and advantages of the invention will be apparent from the following detailed description of illustrative embodiments which proceeds with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] While the appended claims set forth the features of the present invention with particularity, the invention and its advantages are best understood from the following detailed description taken in conjunction with the accompanying drawings, of which:

[0012] Figure 1 is a simplified schematic diagram illustrating an exemplary architecture of a computing device for carrying out the configuration of a computer network in accordance with an embodiment of the invention;

[0013] Figure 2 is a diagram illustrating an arrangement of computing devices for carrying out the configuration of a computer network in accordance with an embodiment of the invention;

[0014] Figure 3 is a simplified schematic illustrating a software architecture for carrying out the configuration of a computer network in accordance with an embodiment of the invention;

[0015] Figure 4 is a flow diagram illustrating a method of generating network settings and storing the settings to a portable media device in accordance with an embodiment of the invention;

[0016] Figures 5A-J are screenshots of a wizard application for generating network settings and storing the settings to a portable media device in accordance with an embodiment of the invention;

[0017] Figure 6 is a flow diagram illustrating a method of generating wireless network settings and storing the network settings on a portable media device;

[0018] Figure 7 is a flow diagram illustrating a method of generating local area network settings and storing the network settings on a portable media device;

[0019] Figure 8 is a flow diagram illustrating a method of generating wide area network settings and storing the network settings on a portable media device;

[0020] Figure 9 is a flow diagram illustrating a method of generating broadband modem settings and storing the network settings on a portable media device;

[0021] Figure 10 is a flow diagram illustrating a method uploading network settings from a portable media device;

[0022] Figure 11 is a data structure diagram representing wireless network configuration settings in accordance with an embodiment of the invention;

[0023] Figure 12 is a data structure diagram representing LAN configuration settings in accordance with an embodiment of the invention;

[0024] Figure 13 is a data structure diagram representing WAN configuration settings in accordance with an embodiment of the invention;

[0025] Figure 14 is a data structure diagram representing broadband modem settings in accordance with an embodiment of the invention; and

[0026] Figure 15 is a data structure diagram representing device configuration settings in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0027] Methods and systems for configuring network devices with a portable media device will now be described with respect to certain embodiments. The skilled artisan will readily appreciate that the methods and systems described herein are merely exemplary and that variations can be made without departing from the spirit and scope of the invention.

[0028] The present invention will be more completely understood through the following detailed description, which should be read in conjunction with the attached drawings. In this description, like numbers refer to similar elements within various embodiments of the present invention. The invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as procedures, being executed by a personal computer. Generally, procedures include program modules, routines, functions, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, and microprocessor-based or programmable consumer electronics devices. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices. The term computer system may be used to refer to a system of computers such as may be found in a distributed computing environment.

[0029] **Figure 1** illustrates an example of a suitable computing system environment 100 in which the invention may be implemented. The computing system environment 100 is only one

example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100. Although at least one embodiment of the invention does include each component illustrated in the exemplary operating environment 100, another more typical embodiment of the invention excludes some or all non-essential components, for example, input/output devices other than those required for network communications.

[0030] With reference to **Figure 1**, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of the computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

[0031] The computer 110 typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer 110 and includes both volatile and nonvolatile media, and removable and nonremovable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and nonremovable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a

manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above are included within the scope of computer-readable media.

[0032] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. By way of example, and not limitation, **Figure 1** illustrates operating system 134, application programs 135, other program modules 136 and program data 137.

[0033] The computer 110 may also include other removable and nonremovable, volatile and nonvolatile computer storage media. By way of example only, **Figure 1** illustrates a hard disk drive 141 that reads from or writes to nonremovable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CDROM. Other computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, DVDs, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a nonremovable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0034] The drives and their associated computer storage media, discussed above and illustrated in **Figure 1**, provide storage of computer-readable instructions, data structures, program modules and other data for the computer 110. In **Figure 1**, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146 and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum, they are different copies. A user may enter commands and information

into the computer 110 through input devices such as a tablet, or electronic digitizer, 164, a microphone 163, a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 by way of an interface, such as a video interface 190. The monitor 191 may also be integrated with a touch-screen panel or the like. Note that the monitor and/or touch screen panel can be physically coupled to a housing in which the computing device 110 is incorporated, such as in a tablet-type personal computer. In addition, computers such as the computing device 110 may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 194 or the like.

[0035] The computer 110 preferably operates or is adaptable to operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a peer device or other network node, and typically includes some or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in **Figure 1**. The logical connections depicted in **Figure 1** include a LAN 171 and a WAN 173, but may also include other networks. For example, in the present invention, the computer 110 may comprise the source machine from which data is being migrated, and the remote computer 180 may comprise the destination machine. Note however that source and destination machines need not be initially connected by a network or otherwise, but instead, data may be migrated by way of any media capable of being written by the source platform and read by the destination platform or platforms. For example, one non-limiting instance of such a medium is a portable flash memory medium, sometimes referred to as a memory “key” or memory “stick.” Other non-limiting examples will be given below.

[0036] When used in a LAN environment, the computer 110 is connectable to the LAN 171 through a network interface or adapter 170. The computer 110 may also include a modem 172 or other means for establishing communications over the WAN 173. The modem 172, which may be internal or external, may be connected to the system bus 121 by way of the user input

interface 160 or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, **Figure 1** illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0037] Turning to **Figure 2**, a simple example of a computing environment usable in implementing an embodiment of the invention is depicted. In the example shown in **Figure 2**, a computer 200 communicates with a LAN 202 by way of a physical connection.

Alternatively, the computer 200 communicates with the LAN 202 by way of WAN or other communications media. In an embodiment of the invention, the computer 200 is not initially in communications with any other device or network at all. In an embodiment of the invention, the computer 200 executes a program that generates network configuration settings. These settings are stored on a portable media device (PM) 204, such as a USB flash drive, Memory Stick, CompactFlash card, SmartMedia card, or other storage device. The computer 200 may directly accept the PM 204 by way of a built-in USB port, but alternatively is connected to a peripheral device, such as a card reader, that accepts the PM 204. In one embodiment, the PM 204 is a Bluetooth device that communicates wirelessly with the computer 200.

[0038] Once settings have been generated and stored on the PM, then, by attaching the PM 204 to various networkable devices, the appropriate network configuration settings are transferred to those devices, enabling network communications over a LAN 202, WAN 206, WLAN 208 or other types of computer networks. Exemplary networkable devices accepting the PM 204 include other computers 210 physically connected to the LAN 202. Although the other computers 210 are physically connected to the same LAN 202, they generally cannot communicate with one another until their network settings are appropriately configured. By attaching the PM 204, the appropriate network settings are transferred to the other computers 210, allowing for network communication between them. Similarly, a printer 212 accepting the PM 204 is configured for communications on the LAN 202, and/or on a wireless network 203, making the printer 212 accessible to the computer 200 and other devices on the LAN 202 and/or wireless network 203. A networkable television 214 and networkable telephone 216

also accept the PM 204 and are thereby configured for communications on the LAN 202 and/or wireless network 203.

[0039] Other network hardware is likewise configured for network communication by attaching the PM 204. A wireless access point 218 accepts the PM 204 and is thus configured to allow communications between devices on the LAN 202 and other appropriately configured wireless devices. For example, if the computer 200 was equipped with wireless networking hardware, then it could connect to the LAN 202 by communicating wirelessly with the wireless access point 218. A modem 220 and router 222 also accept the PM 204 and are thereby configured to allow communication between devices on the LAN 202 and devices connected to a WAN 206 or the Internet 224. Notably, attaching the PM 204 allows for network configuration on devices such as modems 220 and routers 222 that typically do not contain input and output mechanisms such as keyboards and monitors.

[0040] The PM 204 is also used to configure wireless networks. For example, a notebook computer 230 and a tablet computer 232 each accept the PM 204 and are thus appropriately configured for wireless communications over an ad hoc wireless network 234. Other computers 236 accepting the PM 204 are similarly configured to communicate over the ad hoc wireless network 234. In this way, access to the ad hoc network is generally limited to those computers that can physically attach the PM 204. Without the PM 204, a user would have to enter configuration settings – including the network name and any security keys – in order to access the wireless network 234. By attaching the PM 204, a computer is granted access to the wireless network 234 quickly and transparently. The computer networks configured by the PM 204 can be peer-to-peer networks (e.g., unmanaged “workgroups” that do not require a dedicated server) or domain-based networks such as client-server networks.

[0041] Turning attention to **Figure 3**, a software architecture for generating and storing network configuration settings is now described, in accordance with an embodiment of the invention. A Flash Config Wizard 302 is executed on a computer 300 and communicates with the computer through a wireless configuration application programming interface (API) 304 to generate wireless network configuration settings. In the Windows operating system environment, for example, the WZCDLG.DLL library can be used. In one embodiment, the Flash Config Wizard 302 is a standalone application. In another embodiment, the Flash Config

Wizard 302 is integrated into a general network setup application. The Flash Config Wizard 302 outputs XML files to the computer through a wireless provisioning API 306. The Flash Config Wizard 302 further outputs XML files to an attached portable media device, such as a USB flash drive 308. The USB flash drive 308 stores files and applications for use in the network configuration process. An embodiment of the invention includes a network setup application 310 stored in the root of the USB Flash Drive 308 in order to facilitate the configuring of network settings on other devices. The network setup application is stored on the USB flash drive 308 by the Flash Config Wizard 302. When the USB flash drive 308 is attached to another device, that device can run the network setup application 310 to load the relevant network settings from the USB flash drive 308 to the other device.

[0042] In an embodiment of the invention, the USB flash drive 308 further stores an autorun file 318. When the USB flash drive 308 is attached to a compatible device that recognizes the autorun file, the detection of the autorun file 318 automatically triggers the device to execute the network setup program 310. In this way, no user intervention is required to transfer the network settings to the device after the USB flash drive 308 has been attached.

[0043] The USB flash drive 308 also stores, in one embodiment, a Flash Config Wizard copy 320. When the USB flash drive 308 is attached to another computer, the Flash Config Wizard 320 is executed by the computer to allow for manual guidance in transferring and modifying the network configuration settings.

[0044] Also stored on the USB flash drive 308 are several XML files representing the generated network configuration settings. These files are stored in a folder named SMRTNKEY on the USB flash drive 308. An LSETTING.XML file 312 contains settings for a LAN. An NSETTING.XML file 314 contains settings for a WAN. A WSETTING.XML file 316 contains settings for a wireless LAN. An MSETTINGS.XML file 313 contains settings for a broadband modem. These XML files are described in greater detail below. By storing these network configuration settings, a single USB flash drive 308 can be used to configure a variety of devices (such as personal computers, routers, printers, PDAs, and WAPs) to communicate over a variety of types of networks. In another embodiment of the invention, the network configuration files are in binary format instead of XML. The binaries are stored on the USB

flash drive, either alone or with the XML files, allowing devices without XML parsers to read the network configuration settings.

[0045] After the network settings configuration files have been transferred to the USB flash drive 308, the USB flash drive 308 is installed in a configurable device 330. Configurable device 330 must provide support for USB Host Port, USB Mass Storage Devices, file allocation table (FAT) 16/32 file systems, and must include an XML Parser. Additionally access points must support 104 bit wireless encryption protocol (WEP), Infrastructure mode, and IEEE 802.11b protocol. In an embodiment of the invention, configurable device 330 includes a light emitting diode (LED) 334 that blinks three times after the network configuration files are downloaded from the USB flash drive 308. In another embodiment of the invention, the USB flash drive 308 includes an LED 325 the blinks three times after the files have been downloaded to the device 330.

[0046] In an embodiment of the invention, the USB flash drive 308 is also used to store a device configuration file 311 for describing the configuration of a device to which the USB flash drive 308 has been attached. Configurable device 330 further includes a device configuration reporter 332 for generating the device configuration file 311. Each device that is configured using the USB flash drive 308 generates an XML file describing the configuration of that device, and reports any fault conditions. The device then writes the device configuration file 311 to the USB flash drive 308, which is next attached to a PC that uploads the device configuration file 311. The PC uses the device configuration file, for example, in a diagnostic tool to determine why network configuration failed. In another example, the PC obtains a uniform resource locator (URL) of the device manufacturer from the device configuration file 311, and uses the URL to download, through hyper text transfer protocol over secure socket layers (HHTPS), an encryption key to access the device. The device configuration file 311 is stored in the SMRTNKY folder in a subfolder having the same name as the device. The device configuration file 311, described in greater detail below, is named using the last eight bytes of the media access control (MAC) address of the device.

[0047] A method for using a portable media device to generate and store network configuration settings, in accordance with an embodiment of the invention, is now described with reference to **Figure 4**. A user installs a portable media device into a computer at step 410.

An interactive configuration tool, or "Flash Config Wizard", is then executed on the computer at step 415. The Flash Config Wizard interacts with the user to generate a set of network settings at step 420. These network settings include configuration data necessary for configuring the computer for use with a wireless network, LAN, WAN, or broadband modem. The settings are written to the portable media device at step 425, and the user removes the portable media device at step 430. At step 435, the user installs the portable media device in a device that the user wishes to configure. The user removes the portable media device after successful configuration, as indicated by a blinking LED, at step 440. At step 445, if there are more devices to be configured, the user repeats steps 435 and 440 for those devices. The user re-installs the portable media device in the computer executing the Flash Config Wizard at step 450. The Flash Config Wizard uploads the device configuration(s) at step 455. In an embodiment of the invention, the Flash Config Wizard uses the device configuration(s) to perform a diagnostic assessment of any errors or faults that occurred in configuring the devices. At step 460, the user prints a hard copy of the various network settings. By printing a hard copy, the user can more easily configure those network devices that do not accept the personal media device and therefore may need to be configured manually.

[0048] The Flash Config Wizard, as used in an embodiment of the invention, is now described in greater detail with respect to the screenshots shown in **Figures 5A -5J**. Although the figures illustrate a program for setting up a wireless network, a Flash Config Wizard is not limited by the invention to configuring wireless networks; other networks such as wired LANs can similarly be configured using the Flash Config Wizard, in an embodiment of the invention. After an introductory screen shown in **Figure 5A**, the user is prompted for a network name and is presented the option of manually or automatically generating a network key in **Figure 5B**. In the screen shown in **Figure 5C**, the user is prompted for a 5 or 10 character ASCn network key, or a 13 or 16 character hexadecimal network key. If the user selected Wi-Fi Protected Access (WPA) encryption in the previous screen, the user is prompted for a 8-63 character ASCII network key, or a 64 character hexadecimal network key, in the screen shown in **Figure 5D**. **Figure 5E** shows an error screen that appears if the user enters an invalid key.

[0049] In the screen shown in **Figure 5F**, the user is prompted to save the settings to a USB flash drive, or set up each device manually. In the screen shown in **Figure 5G**, the Wizard

prompts the user to insert and identify a USB flash drive. When the user clicks "Next" on this screen, the Wizard stores the generated network settings to the portable media device. When the network settings have been completely stored on the portable media device, the user is presented with another screen shown in **Figure 5H**, prompting the user to remove the portable media device and plug it into all additional computers, printers, or other network devices that are to be in communication over the network. After the user is finished configuring additional devices, the user plugs the portable media device back into the computer and is presented with the screen in **Figure 5I** notifying the user of successful configuration. The Wizard also presents the user with a "Print" button, allowing the user to print a hard copy of the network settings, which allows the user to manually configure those network devices that do not accept the portable media device or that cannot be automatically configured for the network. **Figure 5J** depicts the screen shot presented to the user if the user elects to configure devices manually. The user is prompted to print the settings for manual configuration.

[0050] The operation of the Flash Config Wizard is now described with reference to **Figures 6-9**. **Figure 6** illustrates the operation of the Flash Config Wizard with respect to generating a wireless configuration. At step 610, the Flash Config Wizard prompts the user for a network name. The Flash Config Wizard then gets the wireless configuration settings through a wireless configuration API at step 620. The wireless configuration API is provided by the OS, such as WZCDLG.DLL in the Microsoft Windows® OS. In an alternative embodiment, the user manually enters the wireless configuration settings. At step 630, the Flash Config Wizard generates an XML file containing the wireless configuration settings in accordance with a schema. The Flash Config Wizard transfers the XML file to a wireless provisioning API in the OS so that the host computer may be configured with the wireless settings at step 640. The Flash Config Wizard also transfers the XML file, along with an autorun file, a network setup application, and a copy of the Flash Config Wizard, to a portable media device at step 650. After the user has configured the wireless devices and re-installed the portable media device in the host computer, the Flash Config Wizard uploads device configuration files from the portable media device at step 660. At step 670, the Flash Config Wizard analyzes the device configuration files to determine whether the wireless devices were successfully configured.

[0051] **Figure 7** illustrates the operation of the Flash Config Wizard with respect to generating a LAN configuration. The Flash Config Wizard prompts the user to enter various LAN settings at step 710. These settings include, for example, the workgroup name and IP settings. The settings are described in more detail below. In an alternative embodiment, the Flash Config Wizard acquires the LAN settings through an API provided by the OS. At step 720, the Flash Config Wizard generates an XML file containing the LAN configuration settings in accordance with a schema. The Flash Config Wizard transfers the XML file, along with an autorun file, a network setup application, and a copy of the Flash Config Wizard, to a portable media device at step 730. After the user has configured the LAN devices and re-installed the portable media device in the host computer, the Flash Config Wizard uploads device configuration files from the portable media device at step 740. At step 750, the Flash Config Wizard analyzes the device configuration files to determine whether the LAN devices were successfully configured.

[0052] **Figure 8** illustrates the operation of the Flash Config Wizard with respect to generating a WAN configuration. The Flash Config Wizard prompts the user to enter various WAN settings at step 810. These settings include, for example, the type of WAN and a domain name service (DNS) address. The settings are described in more detail below. In an alternative embodiment, the Flash Config Wizard acquires the WAN settings through an API provided by the OS. At step 820, the Flash Config Wizard generates an XML file containing the WAN configuration settings in accordance with a schema. The Flash Config Wizard transfers the XML file, along with an autorun file, a network setup application, and a copy of the Flash Config Wizard, to a portable media device at step 830. After the user has configured the WAN devices and re-installed the portable media device in the host computer, the Flash Config Wizard uploads device configuration files from the portable media device at step 840. At step 850, the Flash Config Wizard analyzes the device configuration files to determine whether the WAN devices were successfully configured.

[0053] **Figure 9** illustrates the operation of the Flash Config Wizard with respect to generating a broadband modem configuration. The Flash Config Wizard prompts the user to enter various broadband modem settings at step 910. These settings are provided by the user's Internet service provider (ISP) and include, for example, a username and password for

broadband Internet access. The settings are described in more detail below. In an alternative embodiment, the Flash Config Wizard acquires the LAN settings through an API provided by the OS. At step 920, the Flash Config Wizard generates an XML file containing the broadband modem configuration settings in accordance with a schema. The Flash Config Wizard transfers the XML file, along with an autorun file, a network setup application, and a copy of the Flash Config Wizard, to a portable media device at step 930. After the user has configured the devices for use with the broadband modem and re-installed the portable media device in the host computer, the Flash Config Wizard uploads device configuration files from the portable media device at step 940. At step 950, the Flash Config Wizard analyzes the device configuration files to determine whether the devices were successfully configured.

[0054] The procedures described with reference to **Figures 6-9** are combined in the Flash Config Wizard when the user wishes to configure multiple types of network settings (e.g. wireless and WAN settings). In this manner, the configuration files are written to the portable media device simultaneously.

[0055] **Figure 10** illustrates the operation of an exemplary device that is configured using a portable media device. At step 1010, a device detects a portable media device containing network configuration files in a device to be configured. The device uploads the relevant network configuration settings from the portable media device and is provisioned at step 1020. If the device supports an automatic discovery protocol, such as UPnP, a computer on the network will immediately recognize the new device and be notified of its availability. In one embodiment, the uploading is performed automatically, without any user intervention. The device searches the stored XML files for the particular data fields it requires, and uploads the corresponding data from the portable media device. In this way, the device only reads data relevant to its operation. At step 1030, the device generates a device configuration file and transfers the file to the portable media device. The device configuration file includes, for example, the manufacturer and model of the devices, as well as network configuration settings. The device configuration file is explained in more detail below.

[0056] A personal computer generates the network settings and device configuration files according to corresponding XML schemas. In one embodiment of the invention, a personal computer generates network settings and device configuration files for multiple profiles. The

files corresponding to each profile are stored in different profile folders on the portable media device. In this manner, a user may maintain configurations for both a home and an office network on the same portable media device.

[0057] In another embodiment, the network settings files include a time-to-live (TTL) setting, and devices configured using the portable media device are denied access to the network after the prescribed TTL has passed. In this way a device is granted only temporary access to the network. In another embodiment, the TTL field is used in conjunction with an authentication scheme, such as a thumbprint. In this way, authenticated devices are granted permanent access to the network while guests are granted only temporary access as determined by the TTL field.

[0058] **Figure 11** illustrates a conceptualization of the wireless network settings file 1100. Each element of file 1100 represents a wireless configuration setting. Service set identifier (SSID) 1110 is a 1-32 byte string representing the name of the wireless network. SSID 1110 can only occur once. Network Key 1115 is a string that the PC will automatically generate, or alternatively, receive from the PC user. The Network Key 1115 is used for encryption on the wireless network. Authentication Type 1120 indicates the authentication protocol used by the wireless network. Authentication Type 1120 allows a range of possible values, including open, shared, WiFi Protected Access (WPA), WPA Pre-Shared Key (PSK), WPA-none, WPA2, or WPA2 PSK. Encryption Type 1125 indicates the encryption protocol used by the wireless network. Encryption Type 1125 allows a range of possible values, including none, Wireless Encryption Protocol (WEP), Temporal Key Integrity Protocol (TKIP), and Advanced Encryption Standard (AES).

[0059] Type 1130 indicates a connection type, and may have as its value either extended service set (ESS) in the case of an ad hoc network, or infrastructure basic service set (IBSS) in the case of an infrastructure network. Key Index 1135 indicates the location of the specific key used to encrypt messages, and may have a value of 1, 2, 3, or 4. Key Index 1135 is used with WEP. Key Provided Info 1140 indicates whether a key is provided automatically, and can have a value of either 0 or 1. 802.1X 1145 indicates whether IEEE 802.1X protocol is used on the network, and can have a value of either 0 or 1. 2.4 GHz Channel 1150 indicates which 2.4 GHz Channel, if any, is being used by the wireless network, and can have a value in the range

of 1 to 14. 5 GHz Channel 1155 indicates which 5 GHz channel, if any, is being used by the wireless network, and can have a value of 36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, or 161. WAP Mode 1160 indicates the mode in which the wireless access point is operating. WAP Mode 1160 can have a value of infrastructure, bridge, repeater, or station.

[0060] EAP Method 1165 indicates the Extensible Authentication Protocol used, and may have a value of EAP-TLS, PEAP-EAP-MSCHAPv2, or PEAP-EAP-TLS. This setting is used for 802.1x authentication only. TTL 1170 indicates a time-to-live element for specifying the length of time a network key is valid.

[0061] **Figure 12** illustrates a conceptualization of a WAN network settings file for WAN configurations. The WAN network settings file 1200 includes the following elements: DNS IP Auto Pushed 1212, Clone MAC Address 1213, MAC Address 1214, DNS1 IP Address 1215, DNS2 IP Address 1216, connectionType 1217, wanConnection 1218, DHCP 1210, Static IP 1220, and PPPoE 1230. DNS IP Auto Pushed 1212 indicates whether a domain name service IP address is automatically pushed to the DHCP client. Clone MAC Address 1213 indicates whether the MAC address of the client should be cloned for use in the WAN, and takes a value of 0 or 1. MAC Address 1214 defines the MAC address of the client. DNS1 IP Address 1215 and DNS2 IP Address 1216 define IP addresses for domain name servers. The connectionType element 1217 specifies whether the connection is via Point-to-Point Protocol over Ethernet (PPPoE) or IP. The PPPoE element 1230 further includes subelements Username 1231, Password 1232, Service Name 1233, Max Idle Time 1234, and Auto-Reconnect 1235. Username 1231 defines a username for a device to be configured, and Password 1232 defines a password for a device to be configured. Service Name 1232 defines a string indicating the name of a PPPoE server. Max Idle Time 1234 defines a maximum idle time for the device to be configured. Auto-Reconnect 1235 indicates whether the device to be configured should automatically reconnect to the network, and takes a value of 0 or 1.

[0062] The wanConnection 1218 element specifies whether the WAN connection is a DHCP connection or a static IP connection. DHCP element 1210 further includes subelement Hostname 1211. Hostname 1211 defines a string signifying the name of the DHCP host. Static IP element 1220 further includes subelements IP Address 1221, IP Subnet 1222, IP Subnet Mask 1223, DG 1224, DNS1 IP Address 1225, and DNS2 IP Address 1226. IP Address 1221

defines the static IP address of the device to be configured. IP Subnet 1222 defines the IP subnet of the device to be configured. IP Subnet Mask 1223 defines the IP subnet mask of the device to be configured. DG 1224 defines the default gateway for the device to be configured.

[0063] **Figure 13** illustrates a conceptualization of the LAN settings configuration file for LAN configuration. File 1300 includes the following elements: Workgroup Name 1310, IP Settings 1320, IP Subnet 1330, Network Shares 1340, Castle 1350, and Domain Name 1360. Workgroup Name 1310 defines a workgroup name for the LAN. IP Settings 1320 defines the IP settings for the LAN device. The IP Settings 1320 may include sub elements DHCP 1321 for defining DHCP settings and Static IP 1322 for defining settings using static IP addressing. IP Subnet 1330 defines an IP subnet for the device to be configured, wherein the IP subnet defined by IP Subnet 1330 may be different from the IP subnet defines in IP Settings 1320. Network Shares 1340 defines the file sharing configuration for the LAN which the device to be configured is to join, including network drive mappings. Castle 1350 is a boolean element for indicating whether Castle-type service is enabled. Domain Name 1360 is a string for indicating the domain name for the LAN.

[0064] **Figure 14** illustrates a conceptualization of the broadband modem settings configuration file for broadband modem configuration. The file 1400 includes the following elements: User Name 1410, Password 1420, DHCP 1430, IP Address 1440, Connection Type 1450, Link Modulation 1460, ATM Encapsulation 1470, and Link Type 1480. User Name 1410 and Password 1420 indicate a user name and password for accessing the Internet through a broadband ISP. DHCP 1430 is a boolean element that indicates whether DHCP is used. IP Address 1440 indicates the IP address assigned to the modem. Connection Type 1450 indicates the type of connection, and may have as a value PPP, PPPoE, PPPoA, IPoA, EoA, Dial PPP, and CIP. Link Modulation 1460 indicate the modulation type, and may have as a value VDSL, SDSL, ADSLdmt, ADSLlite, IDSL, HDSL, QPSK, 16QAM, 64QAM, 256QAM, Ethernet, Other, or POTS. ATM Encapsulation 1470 indicates whether ATM encapsulation is LLC or VCMUX. Link Type 1480 indicates whether the link is via DSL, Cable, Ethernet, or POTS.

[0065] **Figure 15** illustrates a conceptualization of the device settings configuration file 1500. File 1500 includes identification (ID) settings 1510, network settings 1520, and wireless settings 1530 for a device. ID settings 1510 relate to information regarding the manufacturer of

the device, the manufacturer's web site, the model name of the device, the web site for the device model, the firmware version of the firmware in the device, the device's name, a description of the device, the status of the device, a workgroup name for the device, and a presentation URL. Network settings 1520 relate to information regarding the IPv4, IPv6, and MAC addresses of the device, the status of the network interface (e.g. active, no signal, authenticating, disabled), the type of media (e.g. 802.3, 1394, 802.11, USB, Bluetooth), the speed of the connection, and whether DHCP is enabled.

[0066] Wireless settings 1530 relate to information regarding the wireless authentication protocols (e.g. open, shared, WPA-NONE, WPA, WPAPSK, WPA2, WPA2PSK) and the wireless encryption protocols (e.g. WEP, TKIP, AES) used by the device. Wireless settings 1530 further includes wireless settings profiles specific to the function of the device. If the device is a wireless client, wireless settings 1530 includes wi-fi client settings 1532. These settings relate to information regarding the regulatory domain of the device, the transmission power of the device, and the network type (e.g. 802.11a/b/g). If the device is a wireless infrastructure device, wireless settings 1530 includes wi-fi infrastructure settings 1534. These settings relate to information regarding the SSID, network type, channel, and signal strength of the wireless infrastructure. If the device is a WAP, wireless settings 1530 includes WAP settings 1536. These settings relate to information regarding the regulatory domain, MAC filter, beacon interval, and network type of the WAP.

[0067] It will be appreciated that an improved system and method for network device configuration have been disclosed herein. In view of the many possible embodiments to which the principles of the present invention may be applied, it should be recognized that the embodiments described herein with respect to the drawing figures are meant to be illustrative only and should not be taken as limiting the scope of the invention. For example, those of skill in the art will recognize that the illustrated embodiments can be modified in arrangement and detail without departing from the spirit of the invention. Although the invention is described in terms of software modules or components, those skilled in the art will recognize that such may be equivalently replaced by hardware components. Therefore, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.